IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/802,199
Applicants	:	Ka Leung LING et al.
Filed	:	16 March 2004
TC/A.U.	:	2182
Confirmation	:	3761
Examiner	:	Aurangzeb HASSAN
Atty. Docket	:	US000192A

Title: CAN MICROCONTROLLER THAT PERMITS CONCURRENT ACCESS TO DIFFERENT SEGMENTS OF A COMMON MEMORY BY BOTH THE PROCESSOR CORE AND THE DMA ENGINE THEREOF

APPEAL BRIEF

U.S. Patent and Trademark Office
Customer Window, Mail Stop **Appeal Brief - Patents**
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

In response to the Office Action dated 15 January 2008, rejecting pending claims 21-42, and in support of the Notice of Appeal filed on 15 April 2008, Applicants hereby respectfully submit this Appeal Brief.

REAL PARTY IN INTEREST

According to an assignment recorded at Reel 019729, Frame 0843, NXP, B.V., owns all of the rights in the above-identified U.S. patent application.

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences related to this application or to any

related application, nor will the disposition of this case affect, or be affected by, any other application directly or indirectly.

STATUS OF CLAIMS

Claims 1-20 are canceled. Claims 21-42 are pending and all stand rejected. Accordingly, the claims on Appeal are claims 21-42.

STATUS OF AMENDMENTS

There are no pending amendments with respect to this application.

SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to a multiprocessor device, and associated memory arrangement, and a method of synchronizing memory access among processors in a multiprocessor device.[1]

Accordingly, as broadly recited in claim 21, a microcontroller (Fig. 3; page 7, lines 30-31) that supports a plurality of message objects (page 5, line 31 – page 6, line 1), comprises: a processor core (FIG. 3 – element 22; page 7, line 35) that runs applications; a module (FIG. 3 – element 77; page 9, lines 3-8) that processes incoming messages; data memory including a first memory segment (FIG. 3 – element 28; page 8, lines 11-13; page 20, line 16 – page 21, line 1) that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40; page 8, lines 21-25; page 20, line 16 – page 21, line 1) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10; page 10, line 30 – page 11, line 12 – page 12, line 7) for configuration and setup of that message object; and a memory interface unit (FIG. 3 – element 30; page 8, lines 15-17) that

1 In the description to follow, citations to various reference numerals, figures, and corresponding text in the specification are provided solely to comply with Patent Office rules. It should be understood that these reference numerals, figures, and text are exemplary in nature, and not in any way limiting of the true scope of the claims. It would therefore be improper to import anything into any of the claims simply on the basis of **exemplary** language that is provided here only under the obligation to satisfy Patent

permits the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19), and that arbitrates access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 19-26) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the same one of the first and second memory segments. See also page 3, lines 22-31.

As broadly recited in claim 22, the multiprocessor device further features the incoming messages including multi-frame, fragmented messages, and the module (FIG. 3 – element 77) automatically assembling the multi-frame, fragmented messages (FIGs. 11-12) (page 15, line 11- page 17, line 1).

As broadly recited in claim 24, the multiprocessor device further features the processor core (FIG. 3 – element 22), the module (FIG. 3 – element 77), and the memory interface unit (FIG. 3 – element 30) being contained on a single integrated circuit chip (FIG. 3 – element 20) (page 7, line 30 – page 9, line 9).

As broadly recited in claim 25, the multiprocessor device further features the first and second memory segments (FIG. 3 – elements 28 and 40) being contained on the integrated circuit chip (FIG. 3 – element 20) (page 7, line 30 – page 9, line 9).

As broadly recited in claim 27, the multiprocessor device further features the memory interface unit (Fig. 3 – element 40) arbitrating access according to an alternate winner policy, wherein a previous loser is designated a current winner (page 21, lines 19-26).

As broadly recited in claim 28, a microcontroller (Fig. 3; page 7, lines 30-31) that supports a plurality of message objects comprises: a processor core (FIG. 3 – element 22; page 7, line 35) that runs applications; a module (Fig. 3 – element 77; page 9, lines 3-8) that processes incoming messages, wherein the processor core (Fig. 3 – element 22) and the module (FIG. 3 – element 77; page 9, lines 3-8) are contained on a single integrated circuit chip (FIG. 3 – element 20; page 7, line 30 – page 9, line 9); data memory including a first memory space (FIG. 3 – elements 28

Office rules for maintaining an Appeal.

and 40; page 8, lines 11-13, 21-25; page 20, line 16 – page 21, line 1) that is located on the integrated circuit chip (FIG. 3 – element 20; page 7, line 30 – page 9, line 9) and a second memory space (FIG. 5; page 10, lines 12-14) that is located off the integrated circuit chip (FIG. 3 – element 20), the first memory space including a first memory segment (FIG. 3 – element 28 page 8, lines 11-13; page 20, line 16 – page 21, line 1) that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40; page 8, lines 21-25; page 20, line 16 – page 21, line 1) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10; page 10, line 30 – page 11, line 12 – page 12, line 7) for configuration and setup of that message object; and a memory interface unit (FIG. 3 – element 30) that permits the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory spaces (FIG. 3 – elements 28 & 40 and FIG. 5; page 21, lines 11-19), that permits the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19), and that arbitrates access to the second memory space and that arbitrates access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40 & FIG. 5; page 21, lines 11-19) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the second memory space (Fig. 5) or to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40). See also page 3, line 33 – page 4, line 14.

As broadly recited in claim 29, the multiprocessor device further features the incoming messages including multi-frame, fragmented messages (FIGs. 11-12), and the module (FIG. 3 – element 77) automatically assembling the multi-frame, fragmented messages) (page 15, line 11- page 17, line 1).

As broadly recited in claim 34, the multiprocessor device further features the memory interface unit (FIG. 3 – element 30) arbitrating access according to an

alternate winner policy, wherein a previous loser is designated a current winner (page 21, lines 19-26).

As broadly recited in claim 35, a method is provided for operating a microcontroller (FIG. 3; page 7, lines 30-31) that supports a plurality of message objects, the microcontroller including a processor core (FIG. 3 – element 22; page 7, line 35) that runs applications and a module (FIG. 3 – element 77; page 9, lines 3-8) that processes incoming messages, wherein the processor (FIG. 3 – element 22) and module (FIG. 3 – element 77) are included in a same integrated circuit chip (FIG. 3 – element 20; page 7, line 30 – page 9, line 9), and further including a data memory including a first memory space (FIG. 3 – elements 28 & 40) that is located on the integrated circuit chip (FIG. 3 – element 20; page 7, line 30 – page 9, line 9) and a second memory space (FIG. 5; page 10, lines 12-14) that is located off the integrated circuit chip (FIG. 3 – element 20;), the first memory space (FIG. 3 – elements 28 & 40) including a first memory segment (FIG. 3 – element 28) that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10; page 10, line 30 – page 11, line 12 – page 12, line 7) for configuration and setup of that message object. The method comprises: permitting the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19); and arbitrating access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 19-26) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the same one of the first and second memory segments.

As broadly recited in claim 36, the method further features the arbitrating access step being performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner (page 21, lines 19-26).

As broadly recited in claim 37, the method further features the arbitrating step

being performed by a memory interface unit (FIG. 3 – element 30) contained in the microcontroller (page 21, lines 19-26).

As broadly recited in claim 38, a method is provided for operating a microcontroller (FIG. 3) that supports a plurality of message objects, the microcontroller including a processor core (FIG. 3 – element 22) that runs applications, a module (FIG. 3 – element 77) that processes incoming messages, and a data memory including a first memory space that is located on an integrated circuit chip (FIG. 3 – element 20) on which the microcontroller and the module (FIG. 3 – element 77) are incorporated, and a second memory space (FIG. 5) that is located off the integrated circuit chip (FIG. 3 – element 20), the first memory space including a first memory segment (FIG. 3 – element 28) that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10) for configuration and setup of that message object. The method comprises: permitting the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory spaces (FIG. 3 – elements 28 & 40 & FIG. 5; page 21, lines 11-19); permitting the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19); arbitrating access to the second memory space (FIG. 5) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the second memory space (FIG. 5) (page 21, lines 19-26); and arbitrating access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the first and second memory segments (FIG. 3 – elements 28 & 40) (page 21, lines 19-26).

As broadly recited in claim 39, the method further features the arbitrating

access step being performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner (page 21, lines 19-26).

As broadly recited in claim 40, the method further features the arbitrating step being performed by a memory interface unit (FIG. 3 – element 30) contained in the microcontroller (page 21, lines 19-26).

As broadly recited in claim 41, a bus station comprises a microcontroller (FIG. 3) that supports a plurality of message objects (page 1, line 8 – page 2, line 17). The microcontroller further comprises: a processor core (FIG. 3 – element 22; page 7, line 35) that runs applications; a module (FIG. 3 – element 77; page 9, lines 3-8) that processes incoming messages; data memory including a first memory segment (FIG. 3 – element 28; page 8, lines 11-13; page 20, line 16 – page 21, line 1) that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40; page 8, lines 21-25; page 20, line 16 – page 21, line 1) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10; page 10, line 30 – page 11, line 12 – page 12, line 7) for configuration and setup of that message object; and a memory interface unit (FIG. 3 – element 30; page 8, lines 15-17) that permits the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19), and that arbitrates access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 19-26) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the same one of the first and second memory segments. See also page 3, lines 22-31.

As broadly recited in claim 42, a bus station comprises a microcontroller that supports a plurality of message objects (page 1, line 8 – page 2, line 17). The microcontroller further comprises: a processor core (FIG. 3 – element 22; page 7, line 35) that runs applications; a module (FIG. 3 – element 77; page 9, lines 3-8) that processes incoming messages; data memory including a first memory segment (FIG. 3 – element 28; page 8, lines 11-13; page 20, line 16 – page 21, line 1) that provides

a plurality of message buffers associated with respective ones of the message objects, and a second memory segment (FIG. 3 – element 40; page 8, lines 21-25; page 20, line 16 – page 21, line 1) that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields (FIGs. 4, 9 & 10; page 10, line 30 – page 11, line 12 – page 12, line 7) for configuration and setup of that message object; and a memory interface unit (FIG. 3 – element 30; page 8, lines 15-17) that permits the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) to concurrently access a different respective one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 11-19), and that arbitrates access to the same one of the first and second memory segments (FIG. 3 – elements 28 & 40; page 21, lines 19-26) when the processor core (FIG. 3 – element 22) and the module (FIG. 3 – element 77) request concurrent access to the same one of the first and second memory segments. See also page 3, lines 22-31.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on Appeal are: (1) the rejection of claims 21-42 under 35 U.S.C. § 102 over Baji U.S. Patent 5,513,374; and (2) the rejection of claims 21-42 on the ground of obviousness-type double patenting over U.S. patents 6,715,001, 6,732,255, 6,498,287, and 6,647,440.

ARGUMENTS

(1) Claims 21-42 Are All Patentable Over Baji

Claim 21

Among other things, the microcontroller of claim 21 supports a plurality of message objects and includes: (1) data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and (2) a memory interface unit that

permits a processor core and a module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

Applicants respectfully submit that Baji does not disclose this combination of features.

At the outset, in the Office Action of 15 January 2008, it appears that the Examiner cannot make up his mind as to what exactly he thinks in Baji corresponds to the recited message objects.

On paragraph 6, lines 2-3, he states that "*instruction requests are the message objects*." Then, on page 5, he states that: "*a message object according to the specification can be considered to be a communication channel over which a complete message, or a succession of messages, can be transmitted (paragraph [0028]) and Baji teaches the channels for messages in figure 1 accordingly*."

With such confusion over what exactly, in Baji is supposed to correspond to the recited message objects, it is not surprising the problems that remainder of the rejection exhibits.

In particular, the microcontroller of claim 21 includes a data memory having a first memory segment that provides a plurality of message buffers associated with respective message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object, and a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

Applicants respectfully submit that Baji does not disclose any microcontroller including a data memory and memory interface unit with these recited features.

The Examiner states that: (1) data memory 1900 and instruction memory 1400 together correspond to the recited first memory segment that provides a plurality of message buffers associated with respective message objects; (2) the registers of FIG. 4B correspond to the second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and (3) parallel arbiter 2100 supposedly corresponds to the memory interface unit.

At the outset, Applicant notes that Baji teaches that **the registers of FIG. 4B – which the Examiner states supposedly correspond to the second memory segment – are internal registers to DMA Controller (DMAC) 3000** – which the Examiner states supposedly corresponds to the module of claim 1. The Examiner also states that DSP core 3500 supposedly corresponds to the recited processor core.

Again, as mentioned above, in claim 1 a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

In that case, it is clear that the parallel arbiter 2100 of Baji absolutely does not permit DSP core 3500 and DMAC 3000 to concurrently access a different respective one of the first and second memory segments - **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000) as cited by the Examiner**. It is also clear that parallel arbiter 2100 of Baji absolutely does not arbitrate access to the same one of the first and second memory segments - **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000)** - when the DSP core 3500 and DMAC 3000 request concurrent access to the same one of the first and second memory segments - **where the second memory segments are the registers of FIG. 4B (internal**

registers to DMAC 3000). Indeed, it does not appear that DSP core 3500 ever requests any access to the second memory segments - **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000)** as alleged by the Examiner.

Therefore, Baji cannot possibly disclose the microcontroller of claim 21!

Furthermore, in claim 21, the first memory segment – which according to the Examiner reads on both data memory 1900 and instruction memory 1400 of Baji – provides a plurality of message buffers associated with each of the message objects. However, the Examiner cites nothing in Baji that discloses that either or both of memories 1400 and 1900 provide a plurality of message buffers associated with each of a plurality of message objects!

Therefore, again, Baji cannot disclose the microcontroller of claim 21.

Furthermore, in the microcontroller of claim 21, the second memory segment provides a plurality of memory-mapped registers for each message object, wherein the memory-mapped registers contain respective ***command/control fields for configuration and setup of that message object***.

Applicants respectfully submit that Baji does not disclose any microcontroller including the recited memory-mapped registers with these features. The Examiner does not identify which of the many registers shown in FIG. 4B of Baji supposedly contain respective command/control fields for configuration and setup of message objects (and again, as noted above, the Examiner does not clearly and consistently identify what are supposed to be the message objects in Baji in the first place).

The Examiner cites col. 5, lines 54-64 of Baji as supposedly disclosing the recited memory-mapped registers with these features. Reproduced below is the text of Baji at col. 5, lines 54-64:

> In this DSP **1100**, virtually all resources, including registers, internal memories, external memories, internal interfaces to peripheral devices, and the like are "memory mapped", meaning that each such resource has a predefined, unique, address. Both the DMAC **3000** and the DSP Core **3500** can independently access almost all memory mapped locations in the external memory **2500**, instruction memory **1400**, peripheral devices **2300** or data memory **1900** as well as most of the internal registers. By memory mapping all DSP resources, the DSP **1100** also makes all memory mapped resources accessible to the host processor **1200**.

Nowhere in this text does it disclose that DMAC 3000 includes a plurality of memory-mapped registers for each of a plurality of message objects, the memory-mapped registers for each message object ***containing respective command/control fields for configuration and setup of that message object***. Indeed, the cited text does even not mention any registers in DMAC 300, any command/control fields or any message objects!

Therefore, Baji cannot disclose the microcontroller of claim 21.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 21 is patentable over the prior art.

Claims 22-27

Claims 22-27 depend from claim 21 and are deemed patentable over the prior art for at least the reasons set forth above with respect to claim 21, and for the following additional reasons.

Claim 22

Among other things, in the microcontroller of claim 22 the incoming messages include multi-frame, fragmented messages, and the module automatically assembles the multi-frame, fragmented messages.

Applicants respectfully submit that Baji does not disclose such features.

The Examiner states that FIGs. 3B-C show the claimed multi-frame fragmented instructions handled by the DMAC.

Applicants respectfully disagree.

FIGs. 3B-C are timing diagrams showing operations of an internal memory page arbiter.

The Examiner fails to explain how or why he believes that these timing operations illustrate that incoming messages include multi-frame, fragmented messages. More particularly, how can the timing diagrams of FIGs. 3B-C illustrate that the module automatically assembles the multi-frame, fragmented messages?

Accordingly for at least this additional reason, claim 22 is deemed patentable over Baji.

Claim 27

Among other things, in the microcontroller of claim 27, the memory interface unit arbitrates access to the first and second memory segments between the processor core and the module according to an alternate winner policy, wherein a previous loser is designated a current winner.

Applicants respectfully submit that Baji does not disclose such a feature.

The Examiner states that Baji does not disclose such a feature at col. 7, lines 1-34.

The Board is respectfully requested to read the cited text.

Applicants are confident that it will be apparent to the Board that:

(1) the cited text teaches that arbitration for DSP Core 3500 and DMAC 3000 are ***independent*** (not "alternating winner") and are based on a specific priority scheme spelled out on col. 7, lines 5-9; and

(2) contrary to the Examiner's statement in the FINAL Office, the cited text does not disclose that only an ***INTITIAL ACCESS*** is granted based on a priority scheme and succeeding accesses are done in order of receipt.

Accordingly for at least this additional reason, claim 27 is deemed patentable over Baji.

Claim 28

Among other things, in similarity to claim 21, microcontroller of claim 28 includes: (1) data memory including a first memory space that is located on the integrated circuit chip and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at

least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and (2) a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory spaces, that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the second memory space and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the second memory space or to the same one of the first and second memory segments.

As explained above with respect to claim 21, Baji does not disclose this combination of features. In particular, if the internal registers of the DMAC 3000 shown in FIG. 4B are supposed to correspond to the second memory segments, then it is apparent that the parallel arbiter 2100 of Baji absolutely does not permit DSP Core 3500 and DMAC 3000 to concurrently access a different respective one of the first and second memory segments **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000)**, or arbitrate access to the same one of the first and second memory segments **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000)** when the DSP Core 3500 and the DMAC 3000 request concurrent access to the to the same one of the first and second memory segments **where the second memory segments are the registers of FIG. 4B (internal registers to DMAC 3000)**.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 28 is patentable over the prior art.

Claims 29-34

Claims 29-34 depend from claim 28 and are deemed patentable over the prior art for at least the reasons set forth above with respect to claim 28. Furthermore,

claims 29 and 34 are also deemed patentable for the reasons set forth above with respect to similar claims 22 and 27, respectively.

Claim 35

Among other things, the method of claim 35 includes permitting a processor core and a module to concurrently access a different respective one of a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object.

As explained above with respect to claims 21 and 28, Baji does not provide such access to any such first and second memory segments.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 35 is patentable over Baji.

Claims 36 and 37

Claims 36 and 37 depend from claim 35 and are deemed patentable over the prior art for at least the reasons set forth above with respect to claim 35. Furthermore, claim 36 is also deemed patentable for the reasons set forth above with respect to similar claim 27.

Claim 38

Among other things, the method of claim 38 includes permitting a processor core and a module to concurrently access a different respective one of a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object.

As explained above with respect to claims 21, 28 and 35, Baji does not provide such access to any such first and second memory segments.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 38 is patentable over Baji.

Claims 39 and 40

Claims 39 and 40 depend from claim 38 and are deemed patentable over the prior art for at least the reasons set forth above with respect to claim 38. Furthermore, claim 39 is also deemed patentable for the reasons set forth above with respect to similar claim 27.

Claim 41

Among other things, the bus station of claim 41 includes a microcontroller that supports a plurality of message objects and includes: (1) data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and (2) a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

As explained above with respect to claim 21, Baji does not include any microcontroller with this combination of features.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 41 is patentable over Baji.

Claim 42

Among other things, the bus system of claim 42 includes a microcontroller that supports a plurality of message objects and includes: (1) data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective

command/control fields for configuration and setup of that message object; and (2) a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

As explained above with respect to claim 21, Baji does not include any microcontroller with this combination of features.

Accordingly, for at least these reasons, Applicants respectfully submit that claim 42 is patentable over Baji.

(2) Claims 21-42 Are All Patentable over U.S. Patents 6,715,001, 6,732,255, 6,498,287, and 6,647,440

Applicants respectfully traverse the obviousness-type double-patenting rejections of claims 21-42 over U.S. patents 6,715,001, 6,732,255, 6,498,287, and 6,647,440.

The Office Action merely states that "broadening would dictate obviousness" and presents a table that identifies which claims of the present application the Examiner believes present a double-patenting situation with respect to which claims of the four cited patents. Applicants respectfully submit that this approach ignores the detailed analysis required under M.P.E.P. § 804(II)(B). Accordingly, Applicants respectfully submit that the obviousness-type double-patenting rejections of claims 21-42 are improper and should be withdrawn.

CONCLUSION

For all of the foregoing reasons, Applicants submit that claims 21-42 are all patentable over the cited prior art. Therefore, Applicants respectfully request that the rejections of claims 21-42 be withdrawn, the claims be allowed, and the application be passed to issue.

If necessary, the Commissioner is hereby authorized in this reply to charge payment or credit any overpayment to Deposit Account No. 50-0238 for any

additional fees required under 37 C.F.R. § 1.16, 37 C.F.R. § 1.17 or 37 C.F.R. § 41.20, particularly extension of time fees or any additional fee required for filing this Appeal Brief.

Respectfully submitted,

VOLENTINE & WHITT

Date: 16 June 2008

By: _____
Kenneth D. Springer
Registration No. 39,843

VOLENTINE & WHITT
11951 Freedom Drive, Suite 1260
Reston, Virginia 20190
Telephone No.: (571) 283-0724
Facsimile No.: (571) 283-0740

CLAIMS APPENDIX

1-20. (Canceled)

21. (Original) A microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

22. (Original) The microcontroller as set forth in claim 21, wherein the incoming messages include multi-frame, fragmented messages, and the module automatically assembles the multi-frame, fragmented messages.

23. (Original) The microcontroller as set forth in claim 21, wherein the module includes the memory-mapped registers.

24. (Original) The microcontroller as set forth in claim 21, wherein the processor core, the module, and the memory interface unit are contained on a single integrated circuit chip.

25. (Original) The microcontroller as set forth in claim 24, wherein the first and second memory segments are contained on the integrated circuit chip.

26. (Original) The microcontroller as set forth in claim 24, wherein the memory interface unit includes two independent arbiters.

27. (Original) The microcontroller as set forth in claim 21, wherein the memory interface unit arbitrates access according to an alternate winner policy, wherein a previous loser is designated a current winner.

28. (Original) A microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages, wherein the processor core and the module are contained on a single integrated circuit chip;

data memory including a first memory space that is located on the integrated circuit chip and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory spaces, that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the second memory space and that arbitrates access to the same one of the first and second memory segments when the processor core and the module

request concurrent access to the second memory space or to the same one of the first and second memory segments.

29. (Original) The microcontroller as set forth in claim 28, wherein the incoming messages include multi-frame, fragmented messages, and the module automatically assembles the multi-frame, fragmented messages.

30. (Original) The microcontroller as set forth in claim 28, wherein the module includes the memory-mapped registers.

31. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit is contained on the single integrated circuit chip.

32. (Original) The microcontroller as set forth in claim 28, wherein the second memory space provides at least a portion of the message buffer memory space.

33. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit includes two independent arbiters dedicated to a respective one of the first and second memory spaces.

34. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit arbitrates access according to an alternate winner policy, wherein a previous loser is designated a current winner.

35. (Previously Presented) A method for operating a microcontroller that supports a plurality of message objects, the microcontroller including a processor core that runs applications and a module that processes incoming messages, wherein the processor and module are included in a same integrated circuit chip, and further including a data memory including a first memory space that is located on the integrated circuit chip and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at

least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object, the method comprising:

permitting the processor core and the module to concurrently access a different respective one of the first and second memory segments; and,

arbitrating access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

36. (Original) The method as set forth in claim 35, wherein the arbitrating access step is performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner.

37. (Previously Presented) The method as set forth in claim 36, wherein the arbitrating step is performed by a memory interface unit contained in the microcontroller.

38. (Previously Presented) A method for operating a microcontroller that supports a plurality of message objects, the microcontroller including a processor core that runs applications, a module that processes incoming messages, and a data memory including a first memory space that is located on an integrated circuit chip on which the microcontroller and the module are incorporated, and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped

registers for each message object containing respective command/control fields for configuration and setup of that message object, the method comprising:

permitting the processor core and the module to concurrently access a different respective one of the first and second memory spaces;

permitting the processor core and the module to concurrently access a different respective one of the first and second memory segments;

arbitrating access to the second memory space when the processor core and the module request concurrent access to the second memory space; and,

arbitrating access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the first and second memory segments.

39. (Original) The method as set forth in claim 38, wherein the arbitrating access step is performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner.

40. (Original) The method as set forth in claim 39, wherein the arbitrating step is performed by a memory interface unit contained in the microcontroller.

41. (Original) A bus station comprising a microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory

segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

42. (Original) A bus system comprising a microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

EVIDENCE APPENDIX

{None}

RELATED PROCEEEDINGS APPENDIX

{None}